DESIGN AND VERIFICATION ANALYSIS OF AVALON INTERRUPT INTERFACE WITH COVERAGE REPORT

Mahesh Kumar Jha, Richa Sinha and Akhilesh Kumar
Department of E&C Engineering, NIT Jamshedpur, Jharkhand, India

ABSTRACT

Design is an accurate representation of the specification and is carried out at each step of the manufacturing process. The process of verification parallels the design creation process. Verification shows the correctness of the design, makes sure about no bugs in the design and its functionality.The important part of verification process is coverage analysis;it gives idea that to what degree the source code of the DUT has been tested. Code coverage measures how thoroughly our tests exercised the "implementation" of the design specification, and not the verification plan. Functional Coverage is User specified and not automatic like code coverage. It is based on design specification/intent and not on implementation. Avalon interfaces are used to easily connect components in an FPGA in order to simplify system design. It defines the various Interfaces roles which are used in both high-speed streaming and memory-mapped applications. The Avalon bus is a simple bus architecture which makes the connection between on-chip processors and peripherals together into a SOPC. It is an interface that specifies the port connections between master and slave components, and also specifies the timing by which these components communicate. The Avalon Interrupt Interface is one of the roles of Avalon Interface, which allows the components to signal the events to other components. In this paper, code coverage analysis in design is concluded using QuestaSim simulator and functional coverage analysis in verification is concluded using Riviera simulator..

KEYWORDS: *Avalon Interrupt Interface, SystemVerilog, System- on-a-Programmable Chip (SOPC), DUT (Design Under Test), Code Coverage, Functional Coverage, VMM (Verification methodological manual), IP (Intellectual Property), VIP (Verification Intellectual Property)*

I. INTRODUCTION

Avalon interface bus is ALTERA's standard interface bus used in SOPC Builder, through which all Avalon compatible cores can easily communicate. SOPC Builder is a powerful system development tool. It enables us to define and generate a complete SOPC in much less time than using traditional, manual integration methods. The Avalon interface family defines interfaces for streaming high-speed data, reading and writing registers and memory, and controlling off-chip devices. These standard interfaces are designed into the components available in Qsys. We can also use these standardized interfaces in our custom components. By using these standard interfaces, we enhance the interoperability of our designs [1]. Avalon Interrupt Interface is one of the interface roles of Avalon Interface which allows components to signal events to other components. The Avalon interface defines a set of signal types, the behavior of these signals, and the types of transfers supported by these signals [2].In this paper, coverage analysis is done for the design and verification of the Avalon Interrupt Interface.

Coverage analysis is used to measure the effectiveness of verification implementation, a quantitative measurement of the testing space. This analysis states the depth to which the DUT source code has been tested. The goal of verification is to ensure that a design behaves correctly in its real environment [5]. The easiest way to measure verification progress is with code coverage. Here we measure how many lines of code have been executed (line coverage), which paths through the code and expressions have been executed (path coverage), which single-bit variables have had the values 0

or 1 (toggle coverage), and which states and transitions in a state machine have been visited (FSM coverage) [5]. Branch Coverage is one of the types of code coverage which reports the true or false of the conditions like if-else and case statement. Functional coverage is a measure of which design features have been exercised by the tests. Functional coverage is tied to the design intent and is sometimes called "specification coverage," while code coverage measures the design implementation [5]. In this paper, the verification analysis is done using VMM methodology.

The VMM methodology facilitates the creation of robust, reusable and scalable verification environments using SystemVerilog. It is the most widely used and proven verification methodology for SystemVerilog [8] [9].

II. FEATURES OF AVALON INTERRUPT INTERFACE

Some of the prominent features of the Avalon Interface are [2]:

- *Separate Address, Data and Control Lines* – Provides the simplest interface to on-chip logic. By using dedicated address and data paths, Avalon peripherals do not need to decode data and address cycles.
- *Up to 128-bit Data Width* – Supports data paths up to 128 bits. The Avalon interface supports data widths that are not an even power of two.
- *Synchronous Operation* – Provides an interface optimized for synchronous, on-chip peripherals. Synchronous operation simplifies the timing behaviour of the Avalon interface, and facilitates integration with high-speed peripherals.
- *Dynamic Bus Sizing* – Handles the details of transferring data between peripherals with different data widths. Avalon peripherals with differing data widths can interface easily with no special design considerations.
- *Simplicity* – Provides an easy-to-understand interface protocol with a short learning curve.
- *Low resource utilization* – Provides an interface architecture that conserves on-chip logic resources.
- *High performance* – Provides performance up to one-transfer-per-clock.

III. AVALON INTERRUPT INTERFACE

Avalon Interrupt interfaces allow slave components to signal events to master components. For example, a DMA controller can interrupt a processor when it has completed a DMA transfer [1].
It has two categories:

3.1 Interrupt Sender

An interrupt sender drives a single interrupt signal to an interrupt receiver. The timing of the irq signal must be synchronous to the rising edge of its associated clock, but has no relationship to any transfer on any other interface. irq must be asserted until the interrupt has been acknowledged on the associated Avalon-MM (Memory Mapped) slave interface. The interrupt receiver typically determines how to respond to the event by reading an interrupt status register from an Avalon-MM slave interface. The mechanism used to acknowledge an interrupt is component specific [1].

3.2 Interrupt Receiver

An interrupt receiver interface receives interrupts from interrupt sender interfaces. Components with an Avalon-MM master interface can include an interrupt receiver to detect interrupts asserted by slave components with interrupt sender interfaces. The interrupt receiver accepts interrupt requests from each interrupt sender as a separate bit [1]. An interrupt receiver interface receives interrupts from interrupt sender interfaces.

IV. PROPOSED DIAGRAM OF AVALON INTERRUPT INTERFACE

4.1 Individual Request

The interrupt receiver expects to see each interrupt request from each interrupt sender as a separate bit and is responsible for determining the relative priority of the interrupts. The Avalon Interrupt Interface with individual request is shown in Figure 1.

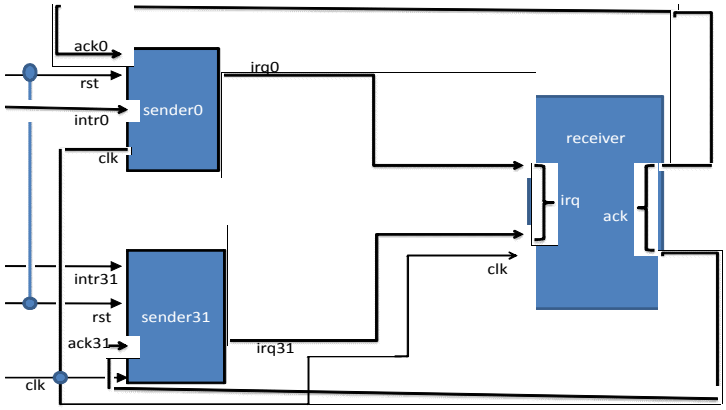

Figure1. Individual request diagram

4.2 Top Level Block Diagram

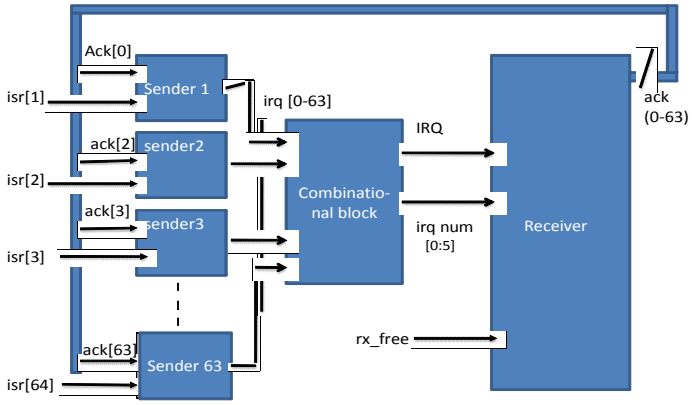

Figure2. Top level Block diagram

The block diagram of Avalon Interrupt Interface is shown in Figure 2. There can be maximum 64 senders.

4.3 Sender Block and Receiver Block

The sender and receiver block of Avalon Interrupt Interface is shown in Figure 3 and Figure 4.The input and output pin description of sender and receiver block has been explained in Table 1.

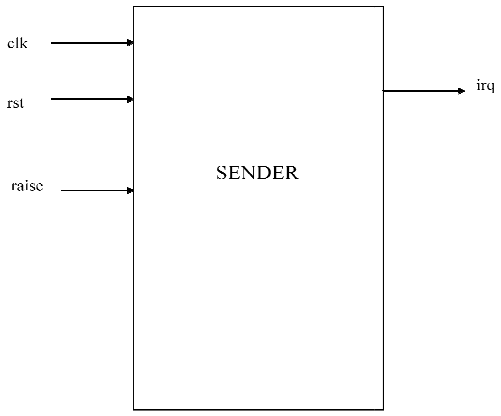

Figure 3. Sender block

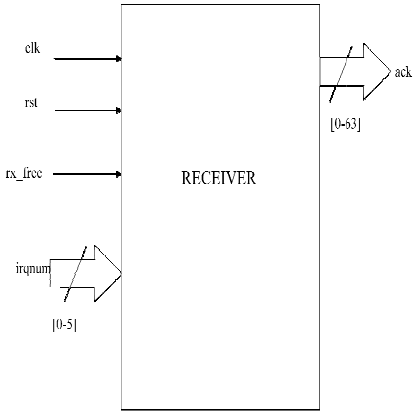

Figure 4. Receiver block

Table 1. Pin Description

Input Pin Description of Sender	
INPUT	**Description**
clk	Clock of 50 MHz
rst	Reset the program
raise	It is a single pulse signal
ack	Acknowledged signal from the Receiver
Output Pin Description of Sender	
OUTPUT	**Description**
irq	Interrupt request
Input Pin Description of Receiver	
INPUT	**Description**
clk	Clock of 50 MHz
rst	Reset the program
rx_en	Enable signal for the Receiver
Irq[63:0]	Interrupt request
Output Pin Description of Receiver	
OUTPUT	**Description**
Ack[63:0]	Acknowledged signal

4.3 State Diagram of Sender and Receiver Block

The state diagram of Sender and Receiver block is shown in Figure 5 and Figure 6.

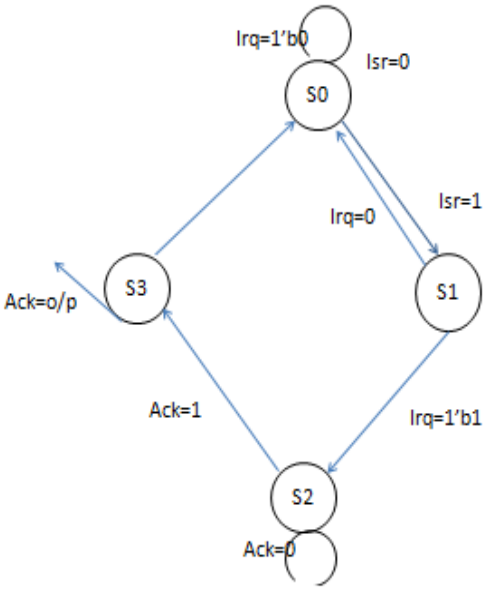

Figure 5. State diagram of Sender

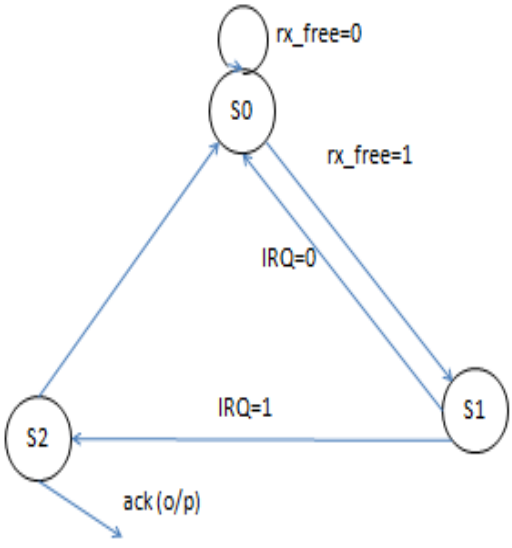

Figure 6. State diagram of Receiver

V. DUT WAVEFORM OF AVALON INTERRUPT INTERFACE

To measure the coverage analysis of design, the code was compiled and then simulated by using Questasim simulator to get the output which is shown in Figure 7.

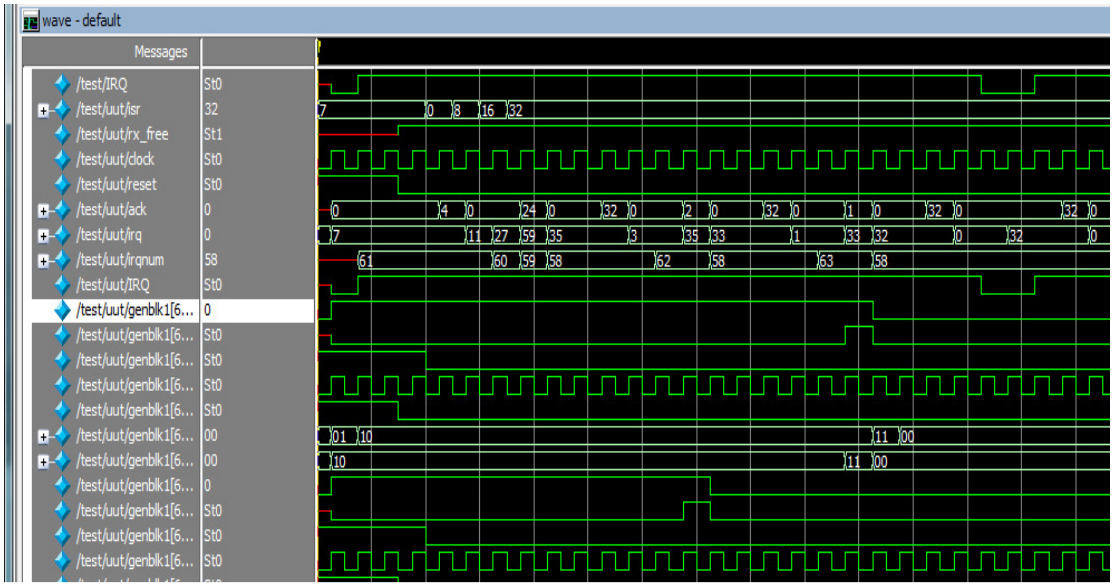

Figure 7. Simulated waveform of Avalon Interrupt Interface

VI. AVALON DUT CODE COVERAGE

Coverage report gives the details of the Code coverage when complete Code analysis was done for the Avalon Interrupt Interface and coverage report as shown in Figure 8.It is found that Global code coverage is 97.10% and Branch coverage is 89.66%.

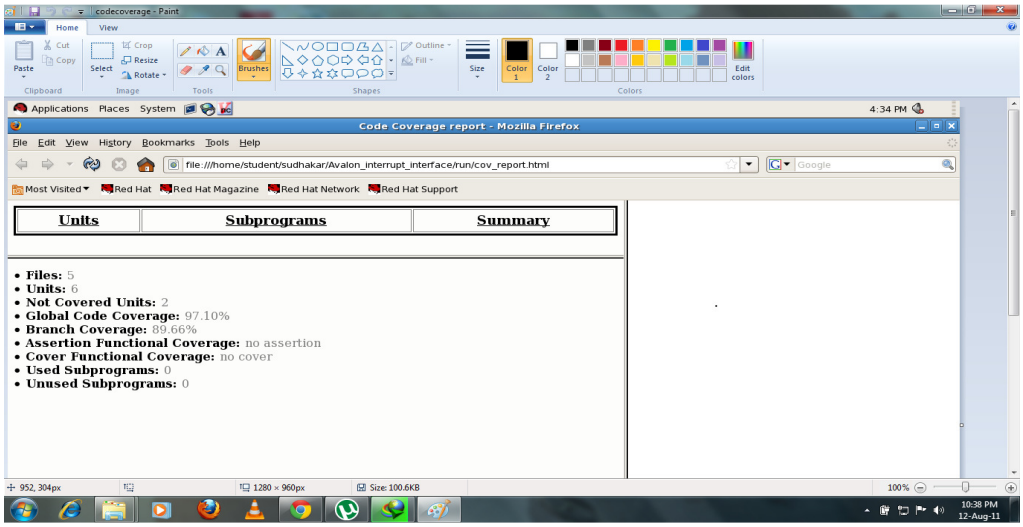

Figure 8. Code coverage analysis

VII. SIMULATION RESULT OF VERIFICATION OF AVALON INTERRUPT INTERFACE

A verification plan is developed to verify the Avalon Interrupt Interface in the VMM environment. The IP is taken as DUT and then it was interfaced with the VMM environment. The VMM environment is created by joining different components written in SystemVerilog coding and IP was written using Verilog coding. The VMM environment was connected to DUT and then compiled using Riviera simulator to get simulated result, which is shown in Figure 9 and Figure 10 in command form.

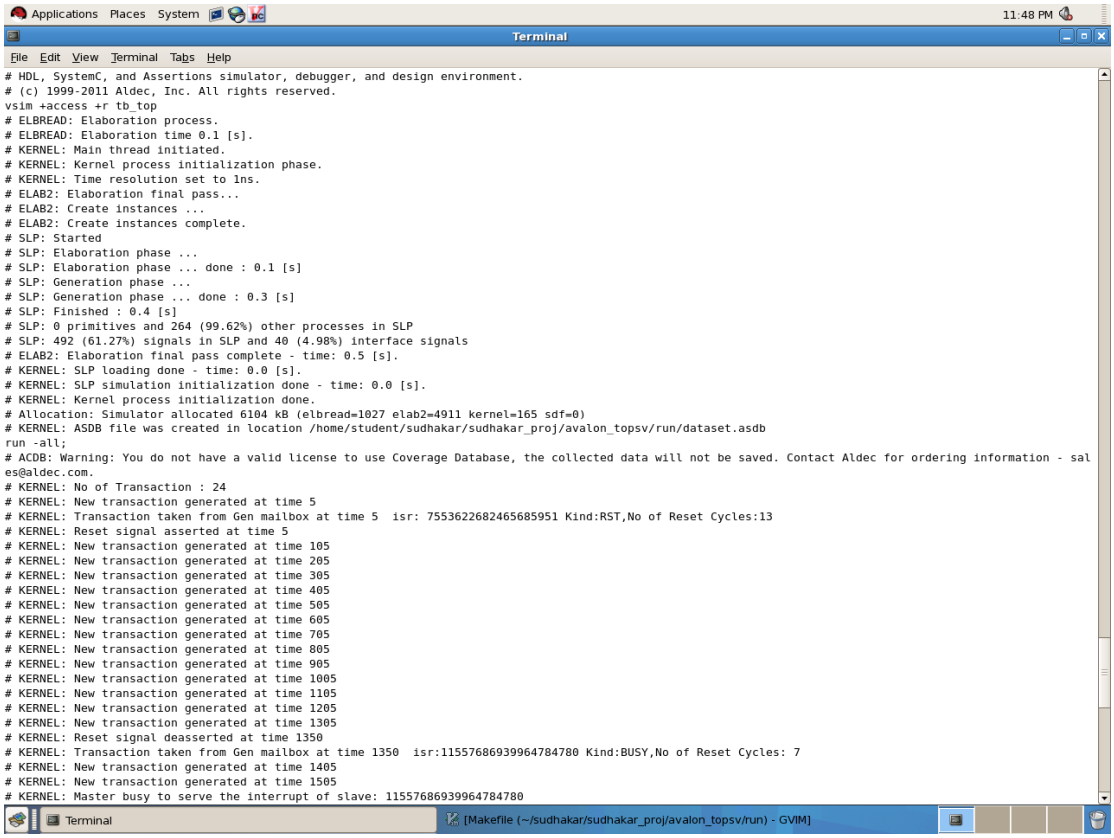

Figure 9. Simulated results of Verification of Avalon interrupt Interface

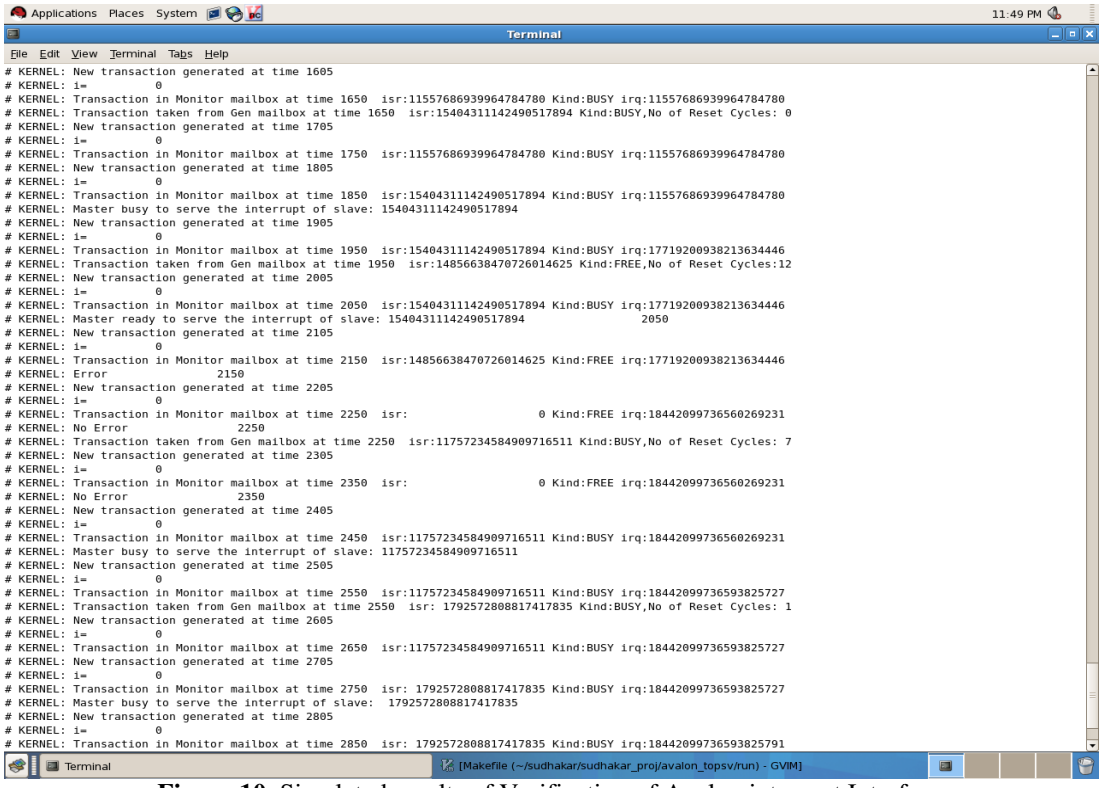

Figure 10. Simulated results of Verification of Avalon interrupt Interface

VIII. COVERAGE ANALYSIS OF VERIFICATION OF AVALON INTERRUPT INTERFACE

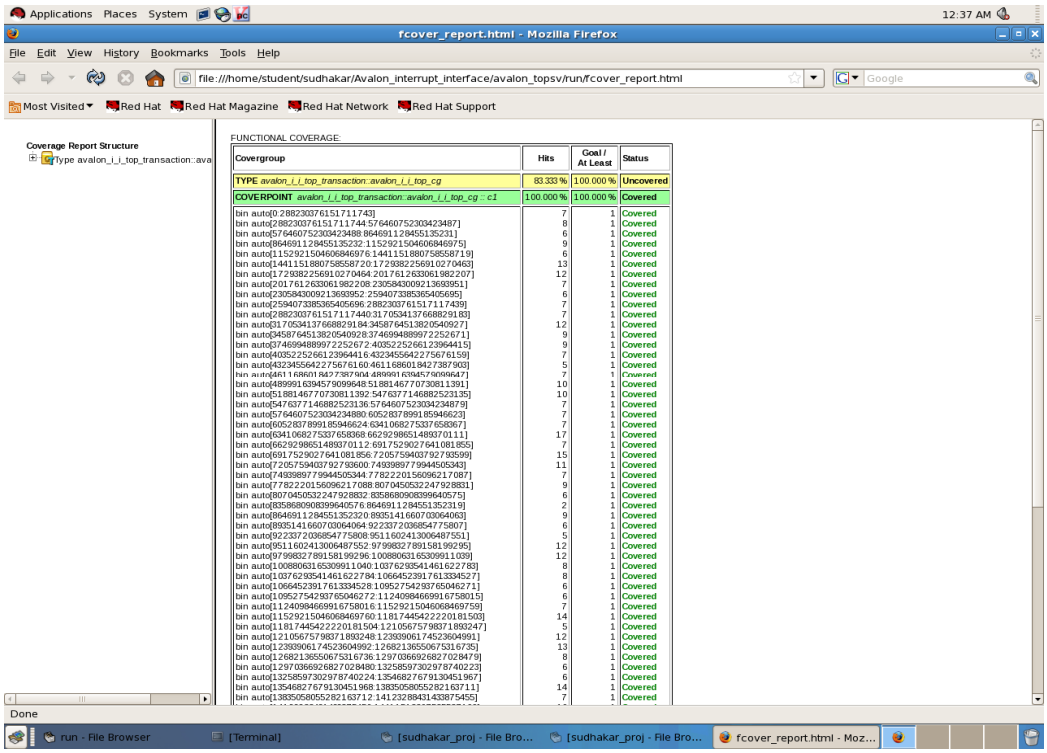

Figure 11. Coverage results of Verification of Avalon interrupt Interface

The Coverage Report gives the details of the functional coverage when complete analysis was done for the DUT and generated coverage report is shown in Figure 6.It is found that the coverage is 100%.

IX. CONCLUSION

The VMM Standard Library provides the foundation base classes for building advanced testbenches, while VMM Application provide higher level function for improved productivity. In this paper Avalon Interrupt Interface IP and VMM complaint VIP is developed. The IP was subjected to various analyses. It was observed that the simulated result of the Avalon Interrupt Interface IP was perfect and code coverage report was also satisfactory. The IP was verified for code coverage using QuestaSim. During the process of code coverage it was found that though the IP gave the expected output, still its overall code coverage is not 100% or approximately nearer to it, indicating that all the paths of the design has not been used to their full strength. So it is clear that higher order exhaustive verification has to be taken to reduce the errors in the design and increase the efficiency of Avalon Interrupt Interface IP. Thus, IP was subjected to VIP developed using VMM environment and was verified exhaustively. The result generated from VIP was found satisfactory and hence Avalon Interrupt interfaces can be used to allow slave components to signal events to master components.

REFERENCES

[1] "Avalon Interface Specifications", Altera Corporation, May 2011.
[2] "Avalon Interface Specification", Altera Corporation, April 2005.
[3] Akhilesh Kumar and Mahesh Kumar Jha, "Coverage Analysis in Verification of Total Zero Decoder of H.264 CAVLD" IJAET Journal, Vol. 1, Issue 5, pp. 197-203, Nov 2011.
[4] Akhilesh Kumar and Chandan Kumar, "Functional Coverage Analysis of OVM Based Verification of H.264 CAVLD Slice Header Decoder" IJAET Journal, Vol. 1, Issue 5, pp. 109-117, Nov 2011.
[5] Chris Spear, SystemVerilog for Verification, New York : Springer, 2006.
[6] Bergeron, Janick. Writing testbenches: functional verification of HDL models. s.l.: Springer, 2003.
[7] Mihaela E.Radhu and Shannon M.Sexton, "Integrating Extensive Functional Verification into digital design Education," IEEE Trans. Educ., vol. 51, no. 3, pp. 385–393, Aug.2008

[8] VMM Golden Reference Guide ,Vers. 1.2, DOULOS, January 2010, Available: www.vmmcentral.org
[9] VMM for SystemVerilog Companion Guide, Vers. 1.1, DOULOS, December 2007, Available: http//www.doulos.com
[10] Samir Palnitkar, "Verilog® HDL: A Guide to Digital Design and Synthesis", Second Edition
[11] http://www.altera.com/support/software/system/sopc/sof-sopc-avalon-interfaces.html
[12] http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.dui0395b/CJHHCEAE.html
[13] www.ict.kth.se/courses/SystemVerilogFunctionalCoverage.pdf
[14] http://www.testbench.co.in

Authors

Mahesh Kumar Jha received B.Tech. Degree from Biju Patnaik University of Technology, Orissa, India in 2007.Currently he is pursuing M. Tech in Department of Electronics and Communication Engineering at N.I.T. Jamshedpur, Jharkhand, India. His interested field of research is VLSI Design.

Richa Sinha received B. E. Degree from RajaRamBapu Institute of Technology Shivaji University, Kolhapur, Maharashtra, India in 2007. Currently she is pursuing M. Tech project work under the guidance of Prof. Akhilesh Kumar in the Department of Electronics & Communication Engg, N. I. T., Jamshedpur. Her interest of field is ASIC Design &Verification.

Akhilesh Kumar received B.Tech degree from Bhagalpur University, Bihar, India in 1986 and M.Tech degree from Ranchi, Bihar, India in 1993. He has been working in teaching and research profession since 1989. He is now working as H.O.D. in Department of Electronics and Communication Engineering at N.I.T. Jamshedpur, Jharkhand, India. His interest of field of research is analog and digital circuit design in VLSI.